

Modeling Overview

A *wireframe model* defines an object only by its edges and vertices. A *surface model* is similar to a wireframe model, but defines an object by its visible surfaces, including faces. A *solid model* defines an object in terms of its size, shape, density, and physical properties (weight, volume, center of gravity, etc.). ACIS is a solid modeler, but wireframe and surface models may also be represented in ACIS.

ACIS *separately* represents the geometry (detailed shape) and the topology (connectivity) of objects. This concept is called *boundary representation*, or *B-rep*, modeling. This provides the ability to determine whether a position is inside, outside, or on the boundary of a volume (which distinguishes a solid modeler from surface or wireframe modelers).

The ACIS model representation consists of various geometric and topologic *entities*, as well as *attributes* that may be attached to the entities. The model is implemented in C++ using a hierarchy of classes.

Refer to the following chapters for more information:

- Chapter 10 ACIS Concepts*, introduces some of the modeling and math concepts (such as geometry, topology, dimensionality, and object space), needed to understand modeling in ACIS, as well as the ACIS concepts of entities, model objects, and attributes.
- Chapter 11 How ACIS Uses C++*, discusses how ACIS uses key features of C++ and the low level C++ classes ACIS implements to establish a math foundation.
- Chapter 12 Geometry*, describes construction and model geometry of ACIS models.
- Chapter 13 Model Topology*, describes the topology of ACIS models.
- Chapter 14 Curves and Surfaces*, discusses curves and surfaces in ACIS.
- Chapter 15 Model Objects*, describes the C++ class ENTITY and its relationship to other object classes in ACIS models.
- Chapter 16 Tolerance Variables and Units*, describes the tolerance variables in ACIS, how they relate to units, and how applications can share model data.
- Chapter 17 Model Modification*, introduces some of the ACIS model modification techniques.
- Chapter 18 Model Analysis*, describes the model analysis capabilities of ACIS.
- Chapter 19 Creating and Modifying Models*, provides examples in Scheme and C++ that illustrate how to create and modify simple models using ACIS.

Specific ACIS modeling functionality that is implemented in a component, such as Boolean operations or blending, is discussed in the corresponding Component Manual.